String Indexing and Slicing behind the scenes! Diagrams.

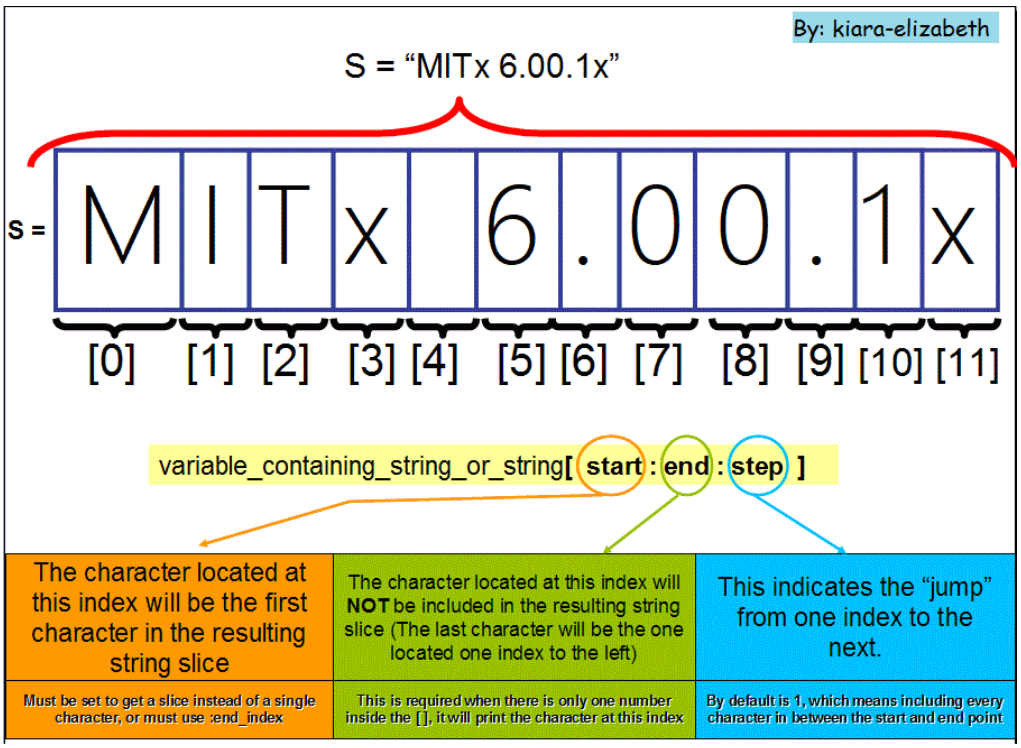
String slicing and indexing are incredibly powerful tools in programming, that's why it's so important to understand **what happens behind the scenes** when you write code and set indices to get a slice or character you're looking for.

As you can see in the diagrams, strings can be thought of as divided by a grid and each character in the string has its own place inside this grid (called **index**, defined by an integer going from 0 upwards for as many characters as the string contains)

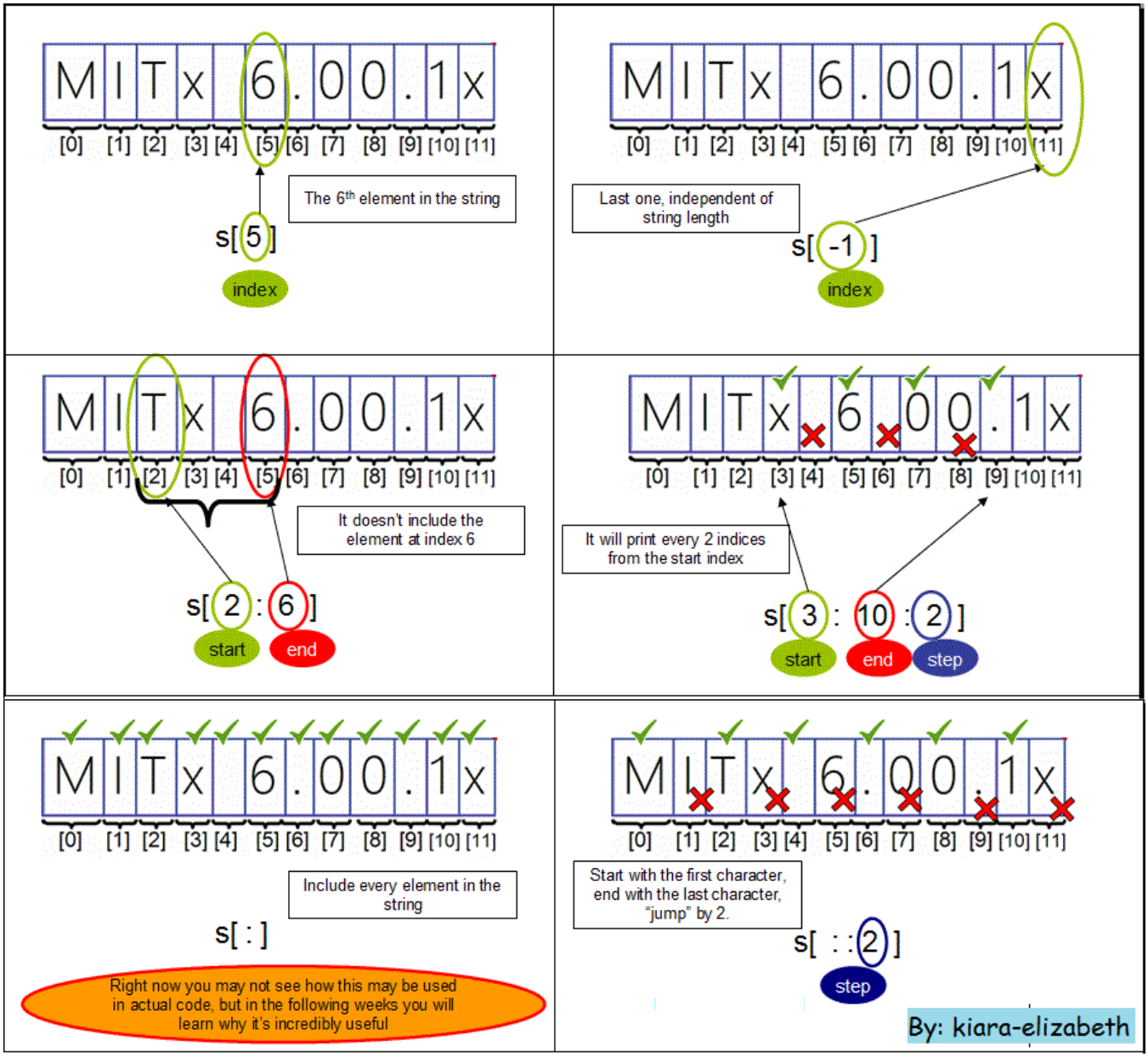
The general format of string slicing is: variable_containing_string_or_string[start: end: step]

Let's look at different alternatives for the parameters inside [ ]:

- If you just set one number inside the **[ ]** e.g 'hi'[1] that number will correspond to the index of a character in the grid, and you will get the single character located at that index.
- If you set a single number but you put a colon (**:**) before the number, like this: **[: (number)]** e.g [:5] you will get a slice of the string that starts from index 0 (this is the default value for the starting index) up to but not including the character located at the index you set as the final index.
- If you set two numbers separated by a colon, like this **[(number1) : (number2)]** e.g [3:5] you will get a slice of the string that starts from 3 and contains all the characters up to but not including the character located at the end index.
- If you set three numbers separated by colons, like this **[(number1) : (number2) : (number3)]** e.g [1:8:2] you will get a slice that starts from the character located at index 1 up to but not including the character located at index 8. And the third parameter determines that characters every two indices will be included. It will jump from index 1 to index 3 and from index 3 to index 5 and so on, without including characters located in between.
- If you set **-1** as the only parameter inside the [ ] , you will get the last character in the string, independent of its length.
- If you set a single number inside [ ] followed by a colon (:) like this: **[(number1) :]** e.g [2:] it will give you a slice of the string that starts at the index you set as the start, and will include every character up to the end of the string.
- If you include two colons (:) followed by a number, like this: **[::(number1)]** that number will be the step ("jump" from one index to the next) e.g [::2].
- If you only set a single colon (:) inside the [ ], you will get a "copy" of the string. In the following weeks, you will learn why this is an important feature and how it can help you in actual code.



Here are some examples of how string slicing works behind the scenes.



Here you can see the result of these examples executed in python's shell:

```
>>> s = 'MITx 6.00.1x'
>>> s[5]
'6'
>>> s[-1]
'x'
>>> s[2:6]
'Tx 6'
>>> s[3:10:2]
'x60.'
>>> s[::2]
'MT .01'
>>> s[:]
'MITx 6.00.1x'
```

Hope it helps! If you have any questions, please share them on the forums!

Estefania (Kiara-elizabeth).